Contents

Getting Envoy's Access Logs

4 minute read ✓ page test

The simplest kind of Istio logging is Envoy’s access logging. Envoy proxies print access information to their standard output. The standard output of Envoy’s containers can then be printed by the `kubectl logs` command.

Before you begin

- Setup Istio by following the instructions in the Installation guide.

The egress gateway and access logging will be enabled if you install the demo configuration profile.

- Deploy the sleep sample app to use as a test source for sending requests. If you have automatic sidecar injection enabled, run the following command to deploy the sample app:

```
$ kubectl apply -f @samples/sleep/sleep.yaml@
```

Otherwise, manually inject the sidecar before deploying the sleep application with the following command:

```
$ kubectl apply -f <(istioctl kube-inject -f @samples/sleep/sleep.yaml
@)
```

> You can use any pod with curl installed as a test source.

- Set the SOURCE_POD environment variable to the name of your source pod:

```
$ export SOURCE_POD=$(kubectl get pod -l app=sleep -o jsonpath={.items
..metadata.name})
```

- Start the httpbin sample.

If you have enabled automatic sidecar injection, deploy the `httpbin` service:

```
$ kubectl apply -f @samples/httpbin/httpbin.yaml@
```

Otherwise, you have to manually inject the sidecar before deploying the `httpbin` application:

```
$ kubectl apply -f <(istioctl kube-inject -f @samples/httpbin/httpbin.
yaml@)
```

Enable Envoy’s access logging

If you used an `IstioOperator` CR to install Istio, add the following field to your configuration:

```
spec:
  meshConfig:
    accessLogFile: /dev/stdout
```

Otherwise, add the equivalent setting to your original `istioctl install` command, for example:

```
$ istioctl install <flags-you-used-to-install-Istio> --set meshConfig.acces
sLogFile=/dev/stdout
```

You can also choose between JSON and text by setting `accessLogEncoding` to `JSON` or `TEXT`.

You may also want to customize the format of the access log by editing `accessLogFormat`.

Refer to global mesh options for more information on all three of these settings:

- `meshConfig.accessLogFile`
- `meshConfig.accessLogEncoding`
- `meshConfig.accessLogFormat`

Default access log format

Istio will use the following default access log format if `accessLogFormat` is not specified:

```
[%START_TIME%] \"%REQ(:METHOD)% %REQ(X-ENVOY-ORIGINAL-PATH?:PATH)% %PROTOCO
L%\" %RESPONSE_CODE% %RESPONSE_FLAGS% %RESPONSE_CODE_DETAILS% %CONNECTION_T
ERMINATION_DETAILS%
\"%UPSTREAM_TRANSPORT_FAILURE_REASON%\" %BYTES_RECEIVED% %BYTES_SENT% %DURA
TION% %RESP(X-ENVOY-UPSTREAM-SERVICE-TIME)% \"%REQ(X-FORWARDED-FOR)%\" \"%R
EQ(USER-AGENT)%\" \"%REQ(X-REQUEST-ID)%\"
\"%REQ(:AUTHORITY)%\" \"%UPSTREAM_HOST%\" %UPSTREAM_CLUSTER% %UPSTREAM_LOCA
L_ADDRESS% %DOWNSTREAM_LOCAL_ADDRESS% %DOWNSTREAM_REMOTE_ADDRESS% %REQUESTE
D_SERVER_NAME% %ROUTE_NAME%\n
```

The following table shows an example using the default access log format for a request sent from `sleep` to `httpbin`:

Log operator	access log

[%START_TIME%]	[2020-11-25T21:26:18.40
\"%REQ(:METHOD)% %REQ(X-ENVOY-ORIGINAL-PATH?:PATH)% %PROTOCOL%\"	"GET /status/418 HTTP/1
%RESPONSE_CODE%	418
%RESPONSE_FLAGS%	-
%RESPONSE_CODE_DETAILS%	via_upstream
%CONNECTION_TERMINATION_DETAILS%	-
\"%UPSTREAM_TRANSPORT_FAILURE_REASON%\"	"-"
%BYTES_RECEIVED%	0

%BYTES_SENT%	135
%DURATION%	4
%RESP(X-ENVOY-UPSTREAM-SERVICE-TIME)%	4
\"%REQ(X-FORWARDED-FOR)%\"	"-"
\"%REQ(USER-AGENT)%\"	"curl/7.73.0-DEV"
\"%REQ(X-REQUEST-ID)%\"	"84961386-6d84-929d-98b
\"%REQ(:AUTHORITY)%\"	"httpbin:8000"
\"%UPSTREAM_HOST%\"	"10.44.1.27:80"
%UPSTREAM_CLUSTER%	outbound|8000||httpbin.

%UPSTREAM_LOCAL_ADDRESS%	10.44.1.23:37652
%DOWNSTREAM_LOCAL_ADDRESS%	10.0.45.184:8000
%DOWNSTREAM_REMOTE_ADDRESS%	10.44.1.23:46520
%REQUESTED_SERVER_NAME%	-
%ROUTE_NAME%	default

Test the access log

1. Send a request from `sleep` to `httpbin`:

   ```
   $ kubectl exec "$SOURCE_POD" -c sleep -- curl -sS -v httpbin:8000/stat
   us/418
   ...
   < HTTP/1.1 418 Unknown
   < server: envoy
   ...
       -=[ teapot ]=-


          _...._
        .'  _ _ `.
       | ."` ^ `". _,
       \_;`"---"`|//
         |       ;/
         \_     _/
           `"""`
   ```

2. Check `sleep`’s log:

```
$ kubectl logs -l app=sleep -c istio-proxy
[2020-11-25T21:26:18.409Z] "GET /status/418 HTTP/1.1" 418 - via_upstre
am - "-" 0 135 4 4 "-" "curl/7.73.0-DEV" "84961386-6d84-929d-98bd-c5ae
e93b5c88" "httpbin:8000" "10.44.1.27:80" outbound|8000||httpbin.foo.sv
c.cluster.local 10.44.1.23:37652 10.0.45.184:8000 10.44.1.23:46520 - d
efault
```

3. Check `httpbin`'s log:

```
$ kubectl logs -l app=httpbin -c istio-proxy
[2020-11-25T21:26:18.409Z] "GET /status/418 HTTP/1.1" 418 - via_upstre
am - "-" 0 135 3 1 "-" "curl/7.73.0-DEV" "84961386-6d84-929d-98bd-c5ae
e93b5c88" "httpbin:8000" "127.0.0.1:80" inbound|8000|| 127.0.0.1:41854
 10.44.1.27:80 10.44.1.23:37652 outbound_.8000_._.httpbin.foo.svc.clus
ter.local default
```

Note that the messages corresponding to the request appear in logs of the Istio proxies of both the source and the destination, `sleep` and `httpbin`, respectively. You can see in the log the HTTP verb (`GET`), the HTTP path (`/status/418`), the response code (`418`) and other request-related information.

Cleanup

Shutdown the sleep and httpbin services:

```
$ kubectl delete -f @samples/sleep/sleep.yaml@
$ kubectl delete -f @samples/httpbin/httpbin.yaml@
```

Disable Envoy's access logging

Remove, or set to `""`, the `meshConfig.accessLogFile` setting in your Istio install configuration.

In the example below, replace `default` with the name of the profile you used when you installed Istio.

```
$ istioctl install --set profile=default
✔ Istio core installed
✔ Istiod installed
✔ Ingress gateways installed
✔ Installation complete
```